

(15.04.04)

# BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

REC'D 20 APR 2004

WIPO

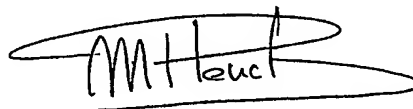
## COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 22 MARS 2004

**PRIORITY DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)

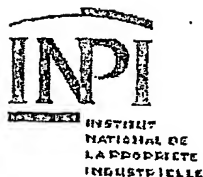
Pour le Directeur général de l'Institut  
national de la propriété industrielle  
Le Chef du Département des brevets



Martine PLANCHE

INSTITUT  
NATIONAL DE  
LA PROPRIÉTÉ  
INDUSTRIELLE

SIEGE  
26 bis, rue de Saint Petersburg  
75800 PARIS cedex 08  
Téléphone : 33 (0)1 53 04 53 04  
Télécopie : 33 (0)1 53 04 45 23  
www.inpi.fr



# BREVET D'INVENTION

## CERTIFICAT D'UTILITE

26bis, rue de Saint-Petersbourg  
75800 Paris Cédex 08  
Téléphone: 01 53.04.53.04 Télécopie: 01.42.94.86.54

Code de la propriété intellectuelle-livre VI

REQUÊTE EN DÉLIVRANCE

DATE DE REMISE DES PIÈCES: 24.06.2003 N° D'ENREGISTREMENT NATIONAL: 0350126 DÉPARTEMENT DE DÉPÔT: 95 DATE DE DÉPÔT: 24.06.2003	SOCIETE CIVILE S.P.I.D. Marion BENETIERE 156 Bd Haussmann 75008 PARIS France
Vos références pour ce dossier: PHFR030043	

<b>1 NATURE DE LA DEMANDE</b>			
Demande de brevet			
<b>2 TITRE DE L'INVENTION</b>			
		PROCÉDE ET SYSTÈME DE TRACE ET DE REMPLISSAGE D'UN PARALLELOGRAMME.	
<b>3 DECLARATION DE PRIORITE OU REQUETE DU BENEFICE DE LA DATE DE DEPOT D'UNE DEMANDE ANTERIEURE FRANCAISE</b>		Pays ou organisation      Date      N°	
<b>4-1 DEMANDEUR</b>			
Nom	KONINKLIJKE PHILIPS ELECTRONICS N.V.		
Rue	Gronewoudseweg 1		
Code postal et ville	5621 BA EINDHOVEN		
Pays	Pays-Bas		
Nationalité	Pays-Bas		
Forme juridique	Société de droit néerlandais		
<b>5A MANDATAIRE</b>			
Nom	SOCIETE CIVILE S.P.I.D.		
Qualité	Liste spéciale: S008, Pouvoir général: 10473		
Affaire suivie par	Marion BENETIERE		
Rue	156 Bd Haussmann		
Code postal et ville	75008 PARIS		
N° de téléphone	0140 76 80 40		
<b>6 DOCUMENTS ET FICHIERS JOINTS</b>			
	Fichier électronique	Pages	Détails
Texte du brevet	textebrevet.pdf	17	D 14, R 2, AB 1
Dessins	dessins.pdf	8	page 8, figures 8
Désignation d'inventeurs			
Pouvoir général			
<b>7 MODE DE PAIEMENT</b>			
Mode de paiement	Prélèvement du compte courant		
Numéro du compte client	001838		

8 RAPPORT DE RECHERCHE				
Etablissement immédiat				
9 REDEVANCES JOINTES	Devise	Taux	Quantité	Montant à payer
062 Dépôt	EURO	0.00	1.00	0.00
063 Rapport de recherche (R.R.)	EURO	320.00	1.00	320.00
Total à acquitter	EURO			320.00

La loi n°78-17 du 6 janvier 1978 relative à l'informatique aux fichiers et aux libertés s'applique aux réponses faites à ce formulaire. Elle garantit un droit d'accès et de rectification pour les données vous concernant auprès de l'INPI.

Signé par

Signataire: FR, Société Civile SPID, A. Bouygues

Emetteur du certificat: DE, D-Trust GmbH, D-Trust for EPO 2.0

Fonction

KONINKLIJKE PHILIPS ELECTRONICS N.V. (Demandeur 1)

## **DESCRIPTION**

### **Domaine technique de l'invention**

La présente invention concerne un procédé de remplissage d'un parallélogramme. Il  
5 concerne également un système mettant en œuvre un tel procédé. Elle concerne enfin un programme d'ordinateur mettant en œuvre un tel procédé.

Elle trouve notamment son application dans les domaines de la synthèse d'images, des jeux vidéo en deux ou trois dimensions et du traitement d'objets vidéo selon la norme MPEG-4.

### **Etat de la technique antérieure**

Dans le domaine des jeux vidéo et plus généralement de la synthèse d'images se pose le problème d'afficher des objets d'intérêt de forme plus ou moins complexe sur un écran graphique comportant une grille discrète de points, communément appelés pixels dans un espace à deux dimensions et voxels dans un espace à trois dimensions. Dans ce but, les  
15 objets d'intérêt sont décomposés en formes primaires, généralement des triangles, qu'il s'agit de tracer et de remplir sur la grille discrète de l'écran à partir de la connaissance des coordonnées de leurs sommets, en leur affectant une valeur de texture. Un problème est donc de faire correspondre à un point d'une forme primaire ayant des coordonnées réelles non entières un point de la grille discrète. Des techniques ont été développées pour y  
20 parvenir, la plupart basés sur les principes d'un algorithme dit « du point milieu » ou algorithme de Bresenham. Cet algorithme, initialement divulgué par le document de Jack E. Bresenham, intitulé « *Algorithm for Computer Control of a Digital Plotter* » et publié dans la revue IBM Systems Journal, volume 4(1), aux pages 25-30 en 1965, permet notamment le tracé de segments obliques de façon rapide et efficace. Un tel algorithme est largement  
25 utilisé dans le domaine de la synthèse d'images. D'autres techniques, elles aussi bien connues de l'homme de métier, ont ensuite été dérivées de cet algorithme pour le remplissage de triangles, voire de polygones.

Ces techniques ont en commun de proposer un remplissage d'un triangle, voire d'un polygone, par balayage horizontal ou vertical. Un type de triangle bien adapté à un tel  
30 balayage est un triangle comprenant un segment droit, c'est-à-dire horizontal ou vertical. La Fig. 1 représente un triangle IJK, dont le segment JK est horizontal. Dans ce cas, le remplissage d'un tel triangle comprend les étapes suivantes :

- une étape de tracé du segment IJ selon l'algorithme de Bresenham. Cette étape est destinée à calculer les coordonnées des points  $J_i$  du segment IJ dans un repère  $(J_x, J_y)$ . Connaissant ces coordonnées, il est ensuite possible de tracer ces points et de leur affecter une valeur de texture,

- une étape de tracé du segment IK selon l'algorithme de Bresenham. Cette étape est destinée à calculer les coordonnées des points  $K_i$  du segment IK dans le repère  $(J_x, J_y)$ ,
- une étape de balayage horizontal du triangle IJK destinée à parcourir une
- une étape de test pour tester si un point de ladite région RI appartient au

5

10

Un avantage de cette méthode est d'être rapide et peu complexe. En effet, l'algorithme de Bresenham peut être mis en œuvre en n'utilisant que des entiers et des additions. Il est donc bien adapté à une mise en œuvre matérielle (en anglais, « hardware »), ce qui est particulièrement intéressant dans le domaine des jeux vidéo. En effet, dans ce domaine, les objets d'intérêt subissent des traitements de plus en plus complexes, qui pour être effectués en temps réel sont mis en œuvre par des processeurs spécialement conçus pour accélérer les calculs réguliers comme par exemple le tracé d'un segment oblique entre deux points connus.

15

20

25

Une telle méthode est aussi avantageusement utilisée dans le domaine de la compression vidéo, notamment par la norme MPEG-4 (de l'anglais, « Motion Picture Expert Group »). En effet, cette norme a pour particularité de considérer un objet d'intérêt d'une séquence vidéo indépendamment du fond de cette séquence. Un tel objet comprend une forme (en anglais, « shape ») et une texture, qui sont encodés séparément. La forme est encodée à l'aide d'une boîte circonscrite (en anglais, « bounding box »), généralement rectangulaire. Le remplissage de cette forme à l'intérieur de la boîte se fait lui aussi à l'aide de l'algorithme de Bresenham.

30

Or, il apparaît que si un objet d'intérêt d'une scène de jeux vidéo peut toujours se décomposer en une mosaïque de triangles, une scène vidéo contient souvent un grand nombre de parallélogrammes. En effet, une forme rectangulaire, très répandue en deux dimensions, devient un parallélogramme dans une vue perspective en trois dimensions.

35

La boîte circonscrite utilisée dans la norme MPEG-4 est soumise aux mêmes types de transformations et devient un parallélogramme dans une vue perspective. Si deux triangles à segment droit suffisent à décrire un rectangle, au moins quatre sont nécessaires pour un parallélogramme. Un parallélogramme est donc représenté par les descriptions de quatre triangles et du chaînage de ces triangles.

Une telle représentation est complexe, coûteuse en espace mémoire et peu pratique à utiliser, lorsqu'il s'agit de tracer et remplir ce parallélogramme sur un écran graphique.

5

### Exposé de l'invention

La présente invention a pour but de proposer une solution pour tracer et remplir un parallélogramme de façon plus simple et plus rapide, à partir des seules coordonnées de ces quatre sommets.

10

Ce but est atteint par un procédé de remplissage d'un parallélogramme comprenant un premier sommet, un deuxième sommet, un troisième sommet et un quatrième sommet, ledit procédé comprenant :

15

- une étape de calcul des coordonnées d'un premier segment entre le premier sommet et le deuxième sommet,
- une étape de calcul des coordonnées d'un deuxième segment entre le premier sommet et le troisième sommet,
- une étape de calcul des coordonnées d'un troisième segment entre le deuxième sommet et le quatrième sommet,
- une étape itérative de calcul des coordonnées d'un segment parallèle au premier segment et compris à l'intérieur du parallélogramme.

20

Le procédé selon l'invention permet de calculer les coordonnées des points situés à l'intérieur d'un parallélogramme à partir des coordonnées de ses quatre sommets, donc de le tracer et de le remplir sans avoir à le décomposer en plusieurs triangles. Ledit procédé consiste à tracer un premier segment du parallélogramme, puis à tracer un deuxième et un troisième segments, parallèles entre eux et situés de part et d'autre du premier segment, et enfin à tracer toutes les parallèles possibles au premier segment contenu entre le deuxième et le troisième segments.

25

Dans un premier mode de réalisation, l'étape itérative de calcul des coordonnées d'un segment parallèle au premier segment consiste à choisir un point du deuxième segment et un point du troisième segment, tels que les deux points soient situés à égale distance respectivement du premier et du deuxième sommet suivant une direction privilégiée horizontale ou verticale et de calculer les coordonnées des points du segment compris entre ces deux points.

30

35

De manière avantageuse, les différentes étapes de calcul des coordonnées des points d'un segment mettent en œuvre l'algorithme de Bresenham, aussi appelé algorithme

du point milieu, car il permet de tracer de façon simple et rapide un segment oblique entre deux points.

En fournissant les coordonnées des points inclus dans le parallélogramme à partir de la seule connaissance de ses quatre sommets, le procédé selon l'invention évite de  
5 décomposer le parallélogramme en triangles et permet d'économiser de l'espace mémoire. En effet, au lieu des sommets de tous les triangles et une représentation du chaînage de ces triangles à l'intérieur du parallélogramme, seules les coordonnées des quatre sommets doivent être stockées en mémoire.

Dans un deuxième mode de réalisation, l'étape itérative de calcul des coordonnées  
10 d'un segment parallèle au premier segment consiste à translater le premier segment selon une direction privilégiée d'un certain incrément à l'intérieur du parallélogramme. Connaissant l'incrément, les coordonnées des points du segment translaté obtenu se déduisent facilement de celles du premier segment. L'opération est renouvelée de façon à couvrir toute la surface du parallélogramme.

Un avantage majeur du deuxième mode de réalisation est de fournir un remplissage  
15 exhaustif du parallélogramme, c'est-à-dire sans aucun trou. Le fait de translater un segment n'entraîne pas d'accumulation d'erreurs. Au contraire, placés côte à côte, les segments translatés s'emboîtent parfaitement, puisqu'ils sont identiques.

## 20 **Brève description des dessins**

Ces aspects de l'invention ainsi que d'autres aspects plus détaillés apparaîtront plus clairement grâce à la description suivante de plusieurs modes de réalisation de l'invention, donnés à titre d'exemples non limitatifs et en regard des dessins annexés parmi lesquels :

- la Fig. 1a illustre le remplissage d'un triangle présentant un bord droit selon l'état de  
25 la technique antérieure,
- la Fig. 1b illustre le remplissage d'un parallélogramme selon l'invention,
- la Fig. 2a présente un schéma fonctionnel du procédé de remplissage d'un parallélogramme selon l'invention,
- la Fig. 2b illustre un exemple de remplissage d'un parallélogramme selon un premier  
30 mode de réalisation de l'invention,
- les Fig. 3a et 3b illustrent le principe de l'algorithme de Bresenham pour le tracé d'un segment oblique sur une grille discrète,
- la Fig. 4 présente un schéma fonctionnel du premier mode de réalisation de l'invention,
- la Fig. 5 illustre la formation de trous lors du remplissage du parallélogramme selon  
35 le premier mode de réalisation de l'invention.

- la Fig. 6 présente un schéma fonctionnel du deuxième mode de réalisation de l'invention.
- les Fig. 7a et 7b illustrent un exemple de remplissage d'un parallélogramme selon un deuxième mode de réalisation de l'invention,
- 5 - la Fig. 8 illustre le principe d'une méthode de calcul des coordonnées du point E selon le deuxième mode de réalisation de l'invention.

### **Exposé détaillé d'au moins un mode de réalisation de l'invention**

10 Le procédé selon l'invention concerne le remplissage d'un parallélogramme sur une grille discrète d'un écran graphique. Ledit procédé s'applique aussi bien à une représentation à deux dimensions qu'à une représentation en trois dimensions. Dans ce qui suit, seul le cas à deux dimensions est décrit. Toutefois, la généralisation au cas à trois dimensions ne posera pas de problème particulier à l'homme du métier.

15 Sur la Fig. 1b, on a représenté un exemple de remplissage d'un parallélogramme ABCD selon l'invention à partir de la seule connaissance de son premier sommet A, de son deuxième sommet B, de son troisième sommet C et de son quatrième sommet D. Le remplissage se fait par balayages successifs du parallélogramme ABCD parallèlement au premier segment AB.

20 Le procédé selon l'invention est présenté de façon fonctionnelle Fig. 2a. Il comprend une étape 10 de calcul des coordonnées des points du premier segment AB. Dans ce qui suit, pour des raisons de simplicité d'écriture, on désignera par AB l'ensemble des coordonnées des points formant le segment AB,  $\{(x_A, y_A), (x_1, y_1), (x_2, y_2), \dots, (x_B, y_B)\}$  dans un repère  $(Ax, Ay)$  d'origine le premier sommet A. L'ensemble formé par les points du premier segment AB s'exprime donc comme suit :  $AB = \{(0,0), (x_1, y_1), (x_2, y_2), \dots, (x_B, y_B)\}$ .

25 Le procédé selon l'invention comprend en outre une étape 11 de calcul des coordonnées des points du deuxième segment AC et une étape 12 de calcul des coordonnées de points du troisième segment BD.

30 De façon avantageuse, ces calcul des coordonnées s'effectuent selon un algorithme dit de Bresenham ou du point milieu. Un tel algorithme permet notamment de tracer un segment oblique sur la grille discrète d'un écran graphique. Le verbe « tracer » signifie ici déterminer des points de la grille discrète formant une approximation discrète du segment oblique.

35 Le principe de l'algorithme de Bresenham est présenté Figs. 3a et 3b. Sur la Fig. 3a, on a représenté une grille discrète 1, repérée par un repère  $(Ox, Oy)$  et coupée par un segment oblique 2. Ledit algorithme procède comme illustré par la Fig. 3b :



- choix d'une direction privilégiée. Ce choix est conditionné par une mesure  $\alpha$  de la pente du segment 2. Si cette pente est inférieure à 45 degrés, l'axe Ox est choisi, sinon c'est l'axe Oy. Dans l'exemple de la Fig. 2b, il s'agit de l'axe Ox,
  - partant d'une extrémité  $S_1$  du segment, incrémentation de la coordonnée  $x_{S1}$  selon l'axe de direction privilégiée Ox. On considère un point S appartenant au segment 2 et tel que  $x_S = x_{S1} + 1$ . Il s'agit de déterminer quel point de la grille 1 est le plus proche de S. Deux points sont candidats : le point  $C_1$  situé au nord est de  $S_1$ , tel que  $x_{C1} = x_{S1} + 1$  et  $y_{C1} = y_{S1} + 1$  et le point E situé à l'est de  $S_1$ , tel que  $x_{C2} = x_{S1} + 1$  et  $y_{C2} = y_{S1}$ .
  - considérant un point M milieu du segment  $C_1C_2$ , on évalue les différences  $\Delta_1 = y_{C1} - y_M$  et  $\Delta_2 = y_M - y_{C2}$ . Si  $\Delta_1 < \Delta_2$ ,  $C_1$  est choisi comme approximation du point S, sinon  $C_2$  est choisi comme approximation du point S,
  - l'opération est répétée pour tous les points du segment oblique 2. Une ligne brisée 3 est obtenue comme une succession de point connexes au sens d'une connexité 8, c'est-à-dire d'une connexité qui considère qu'un point a huit voisins.
- Un exemple d'implémentation d'une fonction « line » destinée à tracer un segment à partir de l'algorithme de Bresenham est présenté ci-dessous en langage C :

```

static
void line(int xi,int yi, int xf,int yf) {
20   int dx,dy,i,xinc,yinc,cumul,x,y ;
      x = xi ;
      y = yi ;
      dx = xf - xi;
      dy = yf - yi;
25   xinc = ( dx > 0 ) ? 1 : -1 ;
      yinc = ( dy > 0 ) ? 1 : -1 ;
      dx = abs(dx) ;
      dy = abs(dy) ;
      printf ("%d, %d\n", x,y) ;
30   if ( dx > dy ) {
      cumul = dx / 2 ;
      for ( i = 1 ; i <= dx ; i++ ) {
      x += xinc ;
      cumul += dy ;
35       if (cumul >= dx) {
          cumul -= dx ;
          y += yinc ;
      }
      printf ("%d, %d\n", x,y) ;
40   }
    }
    else {
      cumul = dy / 2 ;
      for ( i = 1 ; i <= dy ; i++ ) {
45       y += yinc ;
      cumul += dx ;
      if ( cumul >= dy ) {

```

```

        cumul -= dy ;
        x += xinc ;
    }
    printf ("%d, %d\n", x,y) ;
5      }
    }
}

```

10 Cet exemple met en évidence la simplicité de cet algorithme, qui n'utilise que des valeurs entières, aucune multiplication et se résume à quelques lignes de code. Un avantage de l'algorithme de Bresenham est d'être bien adapté à une mise en œuvre matérielle.

Toutefois, l'invention ne se limite pas à l'utilisation de cet algorithme, mais concerne toute autre méthode de tracé de segment oblique. Il existe des méthodes de tracé d'un  
 15 segment qui associent un mot à ce segment, le calcul de ce mot étant basé sur le code de Freeman des connections entre deux points consécutifs de ce segment. Parmi elles, la méthode de Castle est décrite par Castle dans le document « An application of Euclid's algorithm to drawing straight lines », publié dans « Fundamental Algorithms in Computer Graphics » par Springer-Verlag en 1985 aux pages 135 à 139. Un inconvénient de ces  
 20 méthodes est d'être plus complexes à mettre en œuvre que l'algorithme de Bresenham.

La Fig. 2b illustre un exemple de parallélogramme ABCD, dont les premier, deuxième et troisième segments, AB, AC et BD sont tracés à l'aide de l'algorithme de Bresenham. Sur cet  
 25 exemple, le premier segment AB est tracé en prenant l'axe horizontal Ax comme direction privilégiée, tandis que les deuxième et troisième segments AC et BD sont tracés en prenant l'axe vertical Ay comme direction privilégiée.

A l'issue des trois étapes 11, 12 et 13, les tracés des premier, deuxième et troisième segments, AB, AC et BD sont connus. Tous les points de la grille discrète formant ces trois  
 30 segments sont donc parfaitement déterminés par leurs coordonnées dans le repère (Ax, Ay).

Le procédé selon l'invention comprend enfin une étape 20 de tracé de tous les segments parallèles au premier segment AB et compris à l'intérieur du parallélogramme ABCD.

Dans un premier mode de réalisation de l'invention, décrit par le schéma fonctionnel  
 35 de la Fig. 4 et illustré par l'exemple de la Fig. 2b, ladite étape 20 comprend une sous-étape 21 de calcul des coordonnées d'un segment compris entre un point  $A_i$  appartenant au deuxième segment AC et un point  $B_i$  appartenant au troisième segment BD,  $A_i$  et  $B_i$  étant situés à une même distance de A et B respectivement. La sous-étape 21 est itérée afin de tracer tous les segments  $A_i B_i$  situés à l'intérieur du parallélogramme ABCD et à égale  
 40 distance de A et B respectivement selon une direction privilégiée horizontale ou verticale.

Dans l'exemple de la Fig. 2b, la direction privilégiée des segments AC et BD est l'axe vertical Ay. Partant de xA, on considère le point A<sub>1</sub> du deuxième segment AC, tel que  $y_{A_1} = y_A + 1$  et le point correspondant B<sub>1</sub> du troisième segment BD, tel que  $y_{B_1} = y_B + 1$ . Les deuxième et troisième segments AC et BD étant parallèles, les points A<sub>1</sub> et B<sub>1</sub> sont à égale distance de A et B respectivement suivant l'axe Ay.

De façon avantageuse, la sous-étape 21 met en œuvre l'algorithme de Bresenham pour calculer les coordonnées des points du segment A<sub>1</sub>B<sub>1</sub>. L'opération est répétée pour tous les segments A<sub>i</sub>B<sub>i</sub> tels que  $y_{A_i} \leq y_C$ , c'est-à-dire jusqu'à ce que l'intégralité du parallélogramme ait été parcourue.

Un exemple d'implémentation d'une fonction « parallelogram » destinée à remplir un parallélogramme en utilisant la fonction « line » précédemment décrite, est présenté ci-dessous :

```
static void parallelogram(int xa,int ya, int xb,int yb, int xc, int yc, int xd, int yd) {
    int x_ab,y_ab,x_cd,y_cd,i,xinc,yinc,dx,dy,cumul ;

    x_ab = xa ;
    y_ab = ya ;
    x_cd = xc ;
    y_cd = yc ;
    dx = xb - xa ;
    dy = yb - ya ;
    xinc = ( dx > 0 ) ? 1 : -1 ;
    yinc = ( dy > 0 ) ? 1 : -1 ;
    dx = abs(dx) ;
    dy = abs(dy) ;

    line (x_ab, y_ab, x_cd, y_cd);
    printf ("\n");
    if ( dx > dy ) {
        cumul = dx / 2 ;
        for ( i = 1 ; i <= dx ; i++ ) {
            x_ab += xinc ;
            x_cd += xinc ;
            cumul += dy ;
            if (cumul >= dx) {
                cumul -= dx ;
                y_ab += yinc ;
                y_cd += yinc ;
            }
            line (x_ab, y_ab, x_cd, y_cd);
            printf ("\n");
        }
    }
    else {
        cumul = dy / 2 ;
        for ( i = 1 ; i <= dy ; i++ ) {
            y_ab += yinc ;
            y_cd += yinc ;
            cumul += dx ;
```

```

    if ( cumul >= dy ) {
        cumul -= dy ;
        x_ab += xinc ;
        x_cd += xinc ;
5      }
      line (x_ab, y_ab, x_cd, y_cd);
      printf ("\n");
    }
10  }

```

Un avantage de ce premier mode de réalisation de l'invention est d'être très simple. Comme le montre l'exemple de fonction « parallelogram » présenté ci-dessus, l'algorithme de Bresenham n'utilise que des valeurs entières et des opérations peu complexes, limitées à des additions et à des tests comparatifs. Un tel algorithme est donc compatible avec une exécution en temps réel et bien adapté à une mise en œuvre matérielle, comme l'exige la réalisation d'un jeu vidéo par exemple.

Un Inconvénient de ce premier mode de réalisation de l'invention est qu'il ne garantit pas un parcours exhaustif de tous les points de la grille discrète inclus dans le parallélogramme. En effet, l'algorithme de Bresenham est basé sur une approximation, qui engendre une erreur. Sur la Fig. 3b, l'ordonnée  $y_S$  du point S du segment 2 est remplacée par l'ordonnée entière  $y_{C1}$ . L'erreur commise est égale à  $y_{C1} - y_S$ . En générant plusieurs segments obliques côte à côte par l'algorithme de Bresenham, la sous-étape 21 accumule les erreurs de ce type, ce qui entraîne que certains points de la grille discrète, bien qu'inclus dans le parallélogramme, ne soient sur le tracé d'aucun segment  $A_i B_i$ . De telles erreurs sont illustrées par la Fig. 5. Entre le segment tracé entre les points  $A_i$  et  $B_i$  et le segment tracé entre les points A et B se trouvent des points, qui ne sont parcourus par aucun segment du parallélogramme ABCD, bien qu'ils appartiennent au parallélogramme ABCD.

Dans un deuxième mode de réalisation de l'invention, présenté de façon fonctionnelle sur la Fig. 6, l'étape 20 comprend une sous-étape de translation 22 du premier segment AB selon une direction privilégiée horizontale ou verticale, destinée à fournir un segment translaté inclus dans le parallélogramme ABCD. Cette sous-étape de translation 22 consiste à calculer les coordonnées d'un segment parallèle au premier segment AB par translation selon une direction privilégiée horizontale ou verticale. La Fig. 7a présente un exemple de translations horizontales du segment AB selon l'axe  $Ax$  à l'intérieur du parallélogramme ABCD. De telles translations sont très simples à mettre en œuvre. Considérons en effet un segment  $A'_1 B'_1$  translaté horizontalement d'une unité dans le sens des  $x$  positifs. On a simplement  $x_{A'_1} = x_A + 1$  et  $y_{A'_1} = y_A$ . De cette manière, on trace tous les segments translattés  $A'_i B'_i$  possibles, tels que  $x_A < x_{A'_i} \leq x_C$ . Un parallélogramme 6 est formé.

Seuls les points appartenant à une surface commune 5 au parallélogramme 6 avec le parallélogramme ABCD doivent être pris en compte. Pour ce faire, l'étape 20 comprend une sous-étape 23 de test destinée à tester si un point P d'un segment translaté  $A_iB_i$  appartient ou non au parallélogramme ABCD. Ladite étape consiste à vérifier que l'ordonnée  $x_P$  du point

5 P est inférieure à l'ordonnée  $y_{B_i}$  du point  $B_i$  du segment BD tel que  $x_{B_i} = x_B + (x_{A_i} - x_A)$ . Comme le montre la Fig. 7a, de telles translations du premier segment AB selon une direction privilégiée ne permettent pas de remplir l'intégralité du parallélogramme ABCD. Sur la Fig. 7a, considérons le point H d'intersection de l'axe Ax avec le segment CD. Le triangle

10 formé par les sommets ACH reste à remplir. Dans ce but, on considère, en référence à la Fig. 7b, le point E obtenu par projection du troisième sommet C sur le prolongement du segment AB, que l'on désignera par la droite (AB), selon la direction privilégiée Ax. L'étape 20 comprend en outre une sous-étape 24 de calcul des coordonnées dudit point de projection E. Ce calcul peut se faire de différentes manières. Une première solution consiste à considérer

les vecteurs  $\vec{AB} = \begin{pmatrix} x_B - x_A \\ y_B - y_A \end{pmatrix} = \begin{pmatrix} x_B \\ y_B \end{pmatrix}$  et  $\vec{EA} = \begin{pmatrix} x_A - x_E \\ y_A - y_E \end{pmatrix} = \begin{pmatrix} -x_E \\ -y_E \end{pmatrix}$ . Les vecteurs  $\vec{AB}$  et

15  $\vec{EA}$  étant parallèles, l'équation  $-x_B \cdot y_E - y_B \cdot (-x_E) = 0$ , bien connue de l'homme de métier, est vérifiée. Sachant que  $y_E = y_C$ , puisque E est la projection de C selon l'axe Ax, l'abscisse de E vaut  $x_E = (x_B \cdot y_C) / y_B$ . Un avantage de cette solution est d'être basée sur une relation très simple. Un Inconvénient majeur est qu'elle utilise une division, ce qui est très coûteux en termes de cycles de calcul sur un processeur. A titre d'alternative, une méthode plus

20 algorithmique comprenant deux étapes peut être mise en œuvre. Un exemple est présenté Fig. 8. Une première étape consiste à déterminer une position  $E_{NO}$  non optimale de E sur la droite (AB). Elle est décrite comme suit :

$x_{E_{NO}} = y_{E_{NO}} = 0$  ;  
do {  
25  $x_{E_{NO}} += -|x_B - x_A|$  ;  
 $y_{E_{NO}} += -|y_B - y_A|$  ;

}  
while ( $|y_{E_{NO}}| \leq |y_C|$ );

30 On considère le point E' symétrique du point B par rapport au point A. Si ce point E' a une ordonnée supérieure à celle de C en valeur absolue, on prend  $E_{NO}$  égal au point E', sinon on considère le point E'' symétrique de A par rapport à E' et ainsi de suite jusqu'à ce que la condition  $|y_{E_{NO}}| \leq |y_C|$  soit vérifiée.

Une deuxième étape consiste à raffiner la position du point E sur la droite (AB) en déterminant les coordonnées des points du segment  $E_{NO}A$  selon l'algorithme de Bresenham.

Partant de A, il s'agit de parcourir le segment  $AE_{No}$  jusqu'au point E, c'est-à-dire jusqu'au point du segment  $E_{No}A$  d'ordonnée égale à  $y_C$ .

Un avantage de cette deuxième méthode est de ne pas utiliser de division.

5 L'étape 20 comprend ensuite une sous-étape 25 de calcul des coordonnées des points d'un segment EA formé par la projection E et le premier sommet A. De même que les étapes mises en œuvre précédemment pour le tracé de segments obliques, ladite sous-étape 25 met en œuvre de façon avantageuse l'algorithme de Bresenham.

10 Une sous-étape 26 de translation du segment EA selon la direction privilégiée est destinée à fournir un segment  $E_iA_i$  parallèle au segment EA, suivant un principe analogue à celui précédemment décrit pour l'étape 22 de translation du segment AB. De la même manière que précédemment pour le segment AB, la sous-étape 26 est combinée à une sous-étape 27 de test, destinée à tester si un point P d'un segment  $E_iA_i$  translaté de EA est inclus dans le parallélogramme ABCD.

15 Les translations successives des segments AB et EA selon l'axe Ax sur une distance correspondant au segment AC permettent de parcourir tous les points du parallélogramme ABCD. Un avantage de ce deuxième mode de réalisation est de fournir un remplissage intégral du parallélogramme ABCD. On peut considérer que les opérations mises en œuvre pour réaliser une translation d'un segment et tester si ses points sont inclus dans le  
20 parallélogramme ont une complexité équivalente à celles utilisés par l'algorithme de Bresenham pour le tracé d'un segment oblique entre deux points connus. En revanche, l'étape de calcul du point de projection E rend ce deuxième mode de réalisation un peu plus complexe que le premier. Toutefois, comme le montre l'exemple d'implémentation présenté ci-dessous, le deuxième mode de réalisation reste relativement simple à mettre en œuvre.

```

25 void parallelogram(int Ax,int Ay, int Bx,int By,
    int Cx, int Cy, int Dx, int Dy) {
    int AB_dx, AB_dy, AB_max, AB_xinc, AB_yinc,
    AB_cumul, AB_i, AB_x, AB_y, AB_mode;
30 int AC_dx, AC_dy, AC_max, AC_xinc, AC_yinc,
    AC_cumul, AC_i, AC_x, AC_y, AC_mode ;
    int EA_dx, EA_dy, EA_max, EA_xinc, EA_yinc,
    EA_cumul, EA_i, EA_x, EA_y, E_ok, Ex, Ey, EA_mode ;
    int BF_dx, BF_dy, BF_max, BF_xinc, BF_yinc,
35 BF_cumul, BF_i, BF_x, BF_y, F_ok, Fx, Fy, BF_mode ;
    int x, y ;
    int transVertical, transHorizontal ;

    // Déterminer la translation verticale ou horizontale du segment AB
40 transVertical = 0;
    transHorizontal = 0;
    if (abs(Bx - Ax) > abs(By - Ay)) {
        if ( Cy > Ay) transVertical = 1 ;
    }

```

```

else transVertical = -1 ;
}else {
  if ( Cx > Ax) transHorizontal = 1;
  else transHorizontal = -1;
5  }
  //
  // Déterminer la position de E ou F si nécessaire
  // si la translation de AB est horizontale
  // E est la projection horizontale de C sur le segment AB,
  // F est la projection horizontale de D sur le segment AB,
10  // si la translation de AB est verticale
  // E est la projection verticale de C sur le segment AB
  // F est la projection verticale de D sur le segment AB,
  // E et F doivent être à l'extérieur du segment AB
15  if (((transVertical!= 0)&&( Cx < Ax ))||
      ((transHorizontal!= 0)&&( Cy < Ay ))) {
    E_ok = 1;
  }
  else{
20    E_ok = 0;
  }

  if (((transVertical!= 0)&&( Cx > Ax ))||
      ((transHorizontal!= 0)&&( Cy > Ay )))
25    F_ok = 1;
  else
    F_ok = 0;
  AB_x = Ax ;
  AB_y = Ay ;
30  init(Ax, Ay, Bx, By, &AB_dx, &AB_dy, &AB_xinc, &AB_yinc,
      &AB_cumul, &AB_max, &AB_mode);
  for ( AB_i = 1 ; AB_i <= AB_max+1 ; AB_i ++ ) {
    if (transVertical != 0) {
35      if ((F_ok == 1) && (AB_x == Cx)) {
        Fx = Dx;
        Fy = By + (AB_y - Ay);
      }
      if ((E_ok == 1) && (AB_x == Dx)){
40        Ex = Cx;
        Ey = Ay - (By - AB_y);
      }
    }
    }else if ((transHorizontal != 0)){
      if ((F_ok == 1) && (AB_y == Cy)){
45        Fx = Bx + (AB_x - Ax);
        Fy = Dy;
      }
      if ((E_ok == 1) && (AB_y == Dy)){
        Ex = Ax - (Bx - AB_x);
50        Ey = Cy;
      }
    }
  }
  next(AB_mode, AB_xinc, AB_yinc, AB_dx, AB_dy,
      &AB_cumul, &AB_x, &AB_y) ;
55 }

```

```

//
// Bresenham sur le segment AC
AC_x = Ax ;
AC_y = Ay ;
5  init(Ax, Ay, Cx, Cy, &AC_dx, &AC_dy, &AC_xinc, &AC_yinc,
    &AC_cumul, &AC_max, &AC_mode);
for ( AC_i = 1; AC_i <= AC_max+1; AC_i ++ ){
    printf (" \n %dnd line: ", AC_i);
//
10 // Bresenham sur le segment EA
    if ( E_ok == 1 ){
        EA_x = Ex ;
        EA_y = Ey ;
        15  init(Ex, Ey, Ax, Ay, &EA_dx, &EA_dy, &EA_xinc, &EA_yinc,
            &EA_cumul, &EA_max, &EA_mode);
        for ( EA_i = 1 ; EA_i <= EA_max ; EA_i ++ ) {
            x = EA_x;
            y = EA_y;
            20  if ((transVertical == -1))
                y = EA_y - AC_i - transVertical;
            else if (transVertical == 1)
                y = EA_y + AC_i - transVertical;
            if ((transHorizontal == -1))
                x = EA_x - AC_i - transHorizontal;
            25  else if (transHorizontal == 1)
                x = EA_x + AC_i - transHorizontal;
            test(transVertical, transHorizontal,
                AB_dx, AB_dy, AC_x, AC_y, x, y);
            30  next(EA_mode, EA_xinc, EA_yinc, EA_dx, EA_dy,
                &EA_cumul, &EA_x, &EA_y) ;
        }
    }
//
// Bresenham sur le segment AB
35  AB_x = Ax ;
    AB_y = Ay ;
    init(Ax, Ay, Bx, By, &AB_dx, &AB_dy, &AB_xinc, &AB_yinc,
        &AB_cumul, &AB_max, &AB_mode);
    40  for ( AB_i = 1 ; AB_i <= AB_max+1 ; AB_i ++ ) {
        x = AB_x;
        y = AB_y;
        if (transVertical == -1 )
            y = AB_y - AC_i - transVertical;
        else if (transVertical == 1)
            y = AB_y + AC_i - transVertical;
        45  if (transHorizontal == -1)
            x = AB_x - AC_i - transHorizontal;
        else if (transHorizontal == 1)
            x = AB_x + AC_i - transHorizontal;
        50  test(transVertical, transHorizontal,
            AB_dx, AB_dy, AC_x, AC_y, x, y);
        next(AB_mode, AB_xinc, AB_yinc, AB_dx, AB_dy,
            &AB_cumul, &AB_x, &AB_y) ;
    }
55  //

```



```

// Bresenham sur le segment BF
if (F_ok == 1){
    BF_x = Bx ;
    BF_y = By ;
5    Init(Bx, By, Fx, Fy, &BF_dx, &BF_dy, &BF_xinc, &BF_yinc,
        &BF_cumul, &BF_max, &BF_mode);
    for ( BF_i = 1 ; BF_i <= BF_max+1 ; BF_i ++ ) {
        next(BF_mode, BF_xinc, BF_yinc, BF_dx, BF_dy,
            &BF_cumul, &BF_x, &BF_y ) ;
10    x = BF_x;
    y = BF_y;
    if (transVertical == -1)
        y = BF_y - AC_i - transVertical;
    else if (transVertical == 1)
15    y = BF_y + AC_i - transVertical;
    if ((transHorizontal == -1))
        x = BF_x - AC_i - transHorizontal;
    else if (transHorizontal == 1)
        x = BF_x + AC_i - transHorizontal;
20    test(transVertical, transHorizontal,
        AB_dx, AB_dy, AC_x, AC_y, x, y);
    }
}

25    printf ("\n") ;

    next(AC_mode, AC_xinc, AC_yinc, AC_dx, AC_dy,
        &AC_cumul, &AC_x, &AC_y ) ;
30 }
}

```

Il est possible d'implémenter le procédé de traitement selon l'invention au moyen d'un circuit convenablement programmé. Un programme d'ordinateur contenu dans une mémoire de programmation peut amener le circuit à effectuer les différentes opérations décrites précédemment en référence aux Fig. 4 et 6. Le programme d'ordinateur peut aussi être chargé dans la mémoire de programmation par la lecture d'un support de données comme, par exemple, un disque qui contient ledit programme. La lecture peut également s'effectuer par l'intermédiaire d'un réseau de communication comme, par exemple, le réseau Internet. Dans ce cas, un fournisseur de service mettra le programme d'ordinateur à la disposition des intéressés sous la forme d'un signal téléchargeable.

L'invention n'est pas limitée aux modes de réalisation qui viennent d'être décrits à titre d'exemple. Des modifications ou améliorations peuvent y être apportées tout en restant dans le cadre de l'invention. En particulier, d'autres modes d'imagerie, comme l'imagerie par résonance magnétique ou la tomographie par émission de positrons, peuvent être utilisés. Dans le présent texte, le verbe « comprendre » est utilisé pour signifier que l'utilisation d'autres éléments, moyens ou étapes n'est pas exclue.

**REVENDECATIONS**

- 5 1. Procédé de remplissage d'un parallélogramme (ABCD) comprenant un premier sommet (A), un deuxième sommet (B), un troisième sommet (C) et un quatrième sommet (D), ledit procédé comprenant :

  - une étape (10) de calcul des coordonnées des points d'un premier segment (AB) entre le premier sommet (A) et le deuxième sommet (B),
  - une étape (11) de calcul des coordonnées des points d'un deuxième segment (AC) entre le premier sommet (A) et le troisième sommet (C),
  - 10 - une étape (12) de calcul des coordonnées des points d'un troisième segment (BD) entre le deuxième sommet (BD) et le quatrième sommet D,
  - une étape itérative (20) de calcul des coordonnées des points d'un segment parallèle au segment AB et compris à l'intérieur du parallélogramme ABCD.
- 15 2. Procédé de remplissage d'un parallélogramme (ABCD) selon la revendication 1, caractérisé en ce que ladite étape itérative (20) de calcul des coordonnées des points d'un segment parallèle au premier segment (AB) comprend en outre une sous-étape (21) de calcul de coordonnées des points d'un segment compris entre un point ( $A_i$ ) appartenant au deuxième segment (AC) et un point ( $B_i$ ) appartenant au

20 troisième segment (BD), les points ( $A_i$ ) et ( $B_i$ ) étant situés à une même distance du premier sommet (A) et du deuxième sommet (B) respectivement, suivant une direction privilégiée horizontale ( $A_x$ ) ou verticale ( $A_y$ ).
- 25 3. Procédé de remplissage d'un parallélogramme (ABCD) selon la revendication 1, caractérisé en ce que ladite étape itérative (20) de calcul des coordonnées d'un segment parallèle au premier segment AB comprend une sous-étape (22) de translation du premier segment (AB) selon une direction privilégiée horizontale ou verticale, destinée à fournir un segment translaté  $A'_iB'_i$  inclus dans le

30 parallélogramme (ABCD) et une sous-étape (23) de test destinée à tester si un point (P) du segment translaté ( $A'_iB'_i$ ) est inclus dans le parallélogramme ABCD.
- 35 4. Procédé de remplissage d'un parallélogramme (ABCD) selon la revendication 3, caractérisé en ce que ladite étape itérative (20) de calcul des coordonnées des points d'un segment parallèle au premier segment (AB) comprend en outre :

  - une sous-étape (24) de calcul des coordonnées d'une projection (E) du troisième sommet (C) sur un prolongement du premier segment (AB) et parallèlement à la direction privilégiée,

- une sous-étape (25) de calcul des coordonnées des points d'un segment (EA) formé par la projection (E) et le premier sommet (A),
- une sous-étape (26) de translation du segment (EA) selon la direction privilégiée, destinée à fournir un segment (E<sub>i</sub>A'<sub>i</sub>) parallèle au segment (EA),
- 5      - une sous-étape (27) de test destinée à tester si un point (P) d'un segment (E<sub>i</sub>A'<sub>i</sub>) translaté de (EA) est inclus dans le parallélogramme (ABCD).

10      5. Procédé de remplissage d'un parallélogramme (ABCD) selon la revendication 1, caractérisé en ce que lesdites étapes de calcul de coordonnées des points d'un segment mettent en œuvre un algorithme du point milieu.

15      6. Système de tracé et de remplissage d'un parallélogramme (ABCD) pour écran graphique, destiné à fournir les coordonnées de points inclus dans le parallélogramme à partir des coordonnées d'un premier sommet (A), d'un deuxième sommet (B), d'un troisième sommet (C) et d'un quatrième sommet (D), ledit système comprenant :

- des moyens (10) de calcul des coordonnées des points d'un premier segment entre le premier sommet A et le deuxième sommet B,
- 20      - des moyens (11) de calcul des coordonnées des points d'un deuxième segment entre le premier sommet A et le troisième sommet C,
- des moyens (12) de calcul des coordonnées des points d'un troisième segment BD entre le deuxième sommet B et le quatrième sommet D,
- 25      - des moyens itératifs (20) de calcul des coordonnées des points d'un segment parallèle au segment AB et compris à l'intérieur du parallélogramme ABCD.

7. Produit programme d'ordinateur comprenant un jeu d'instructions qui, lorsqu'elles sont chargées dans un circuit, amène celui-ci à effectuer le procédé selon la revendication 1.

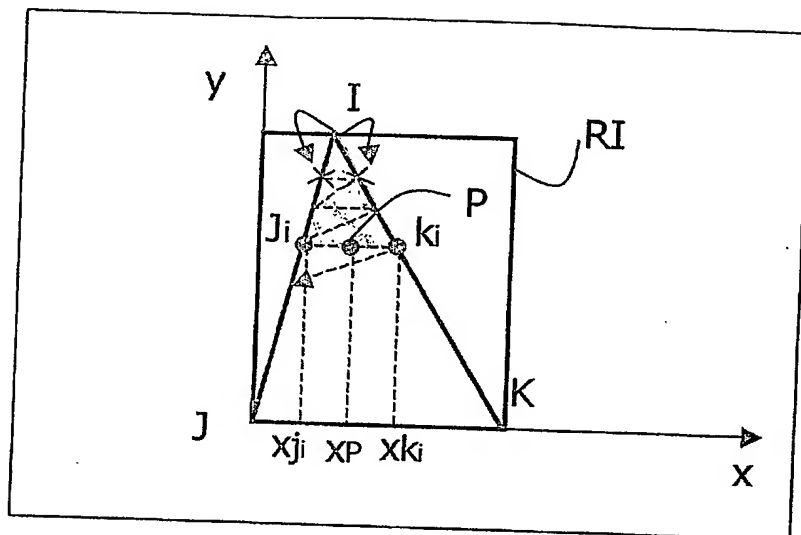


FIG. 1a

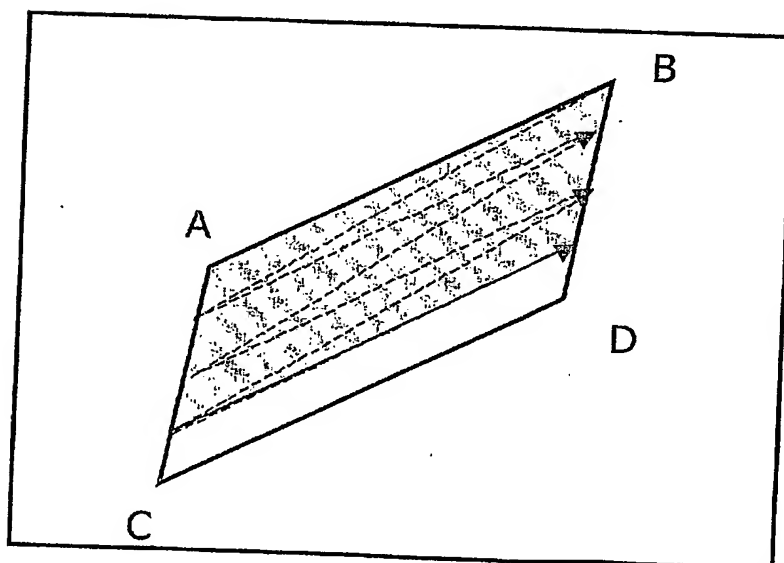


FIG. 1b

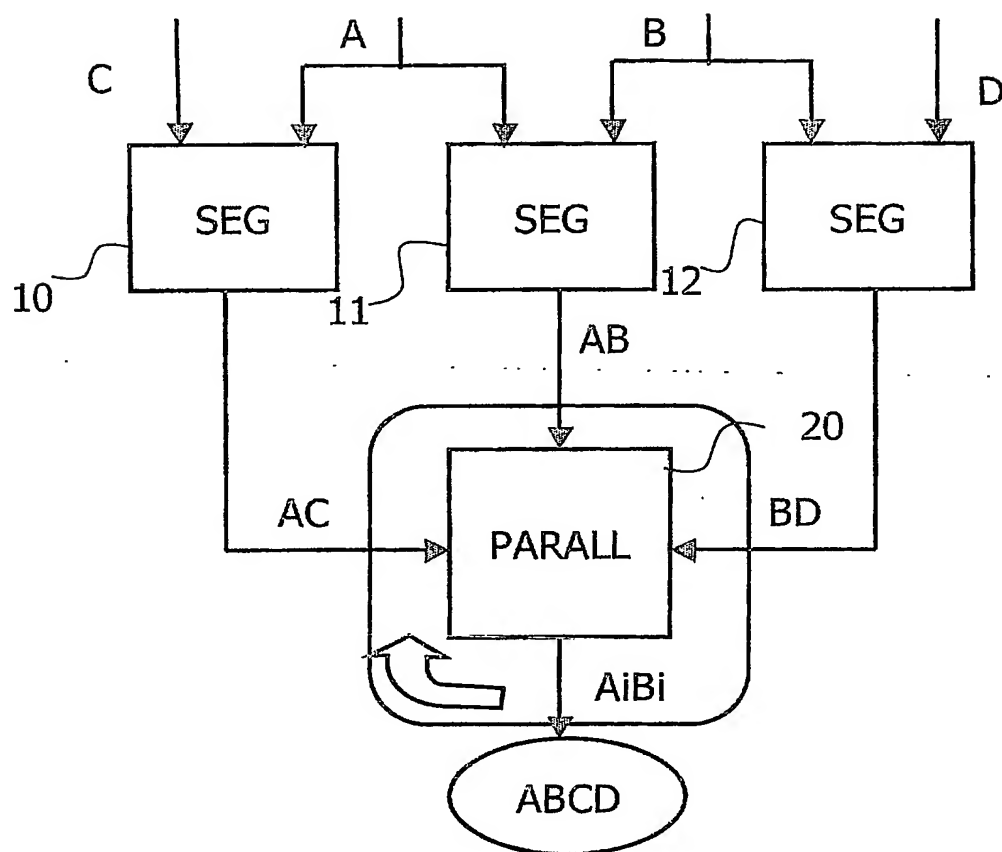


FIG. 2a

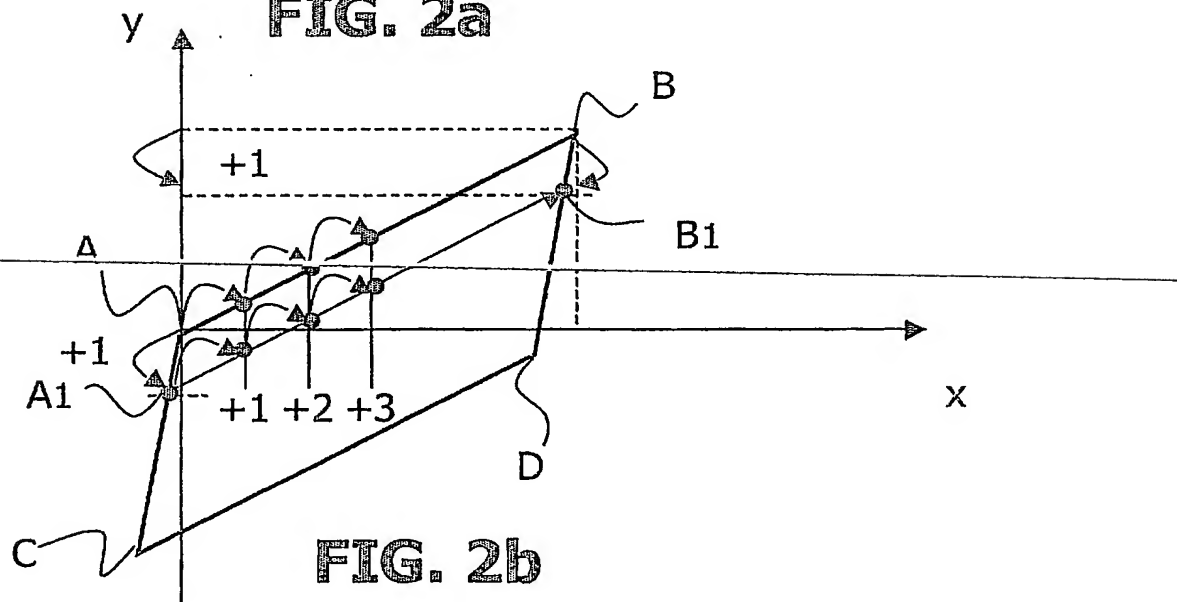
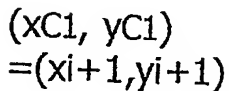


FIG. 2b

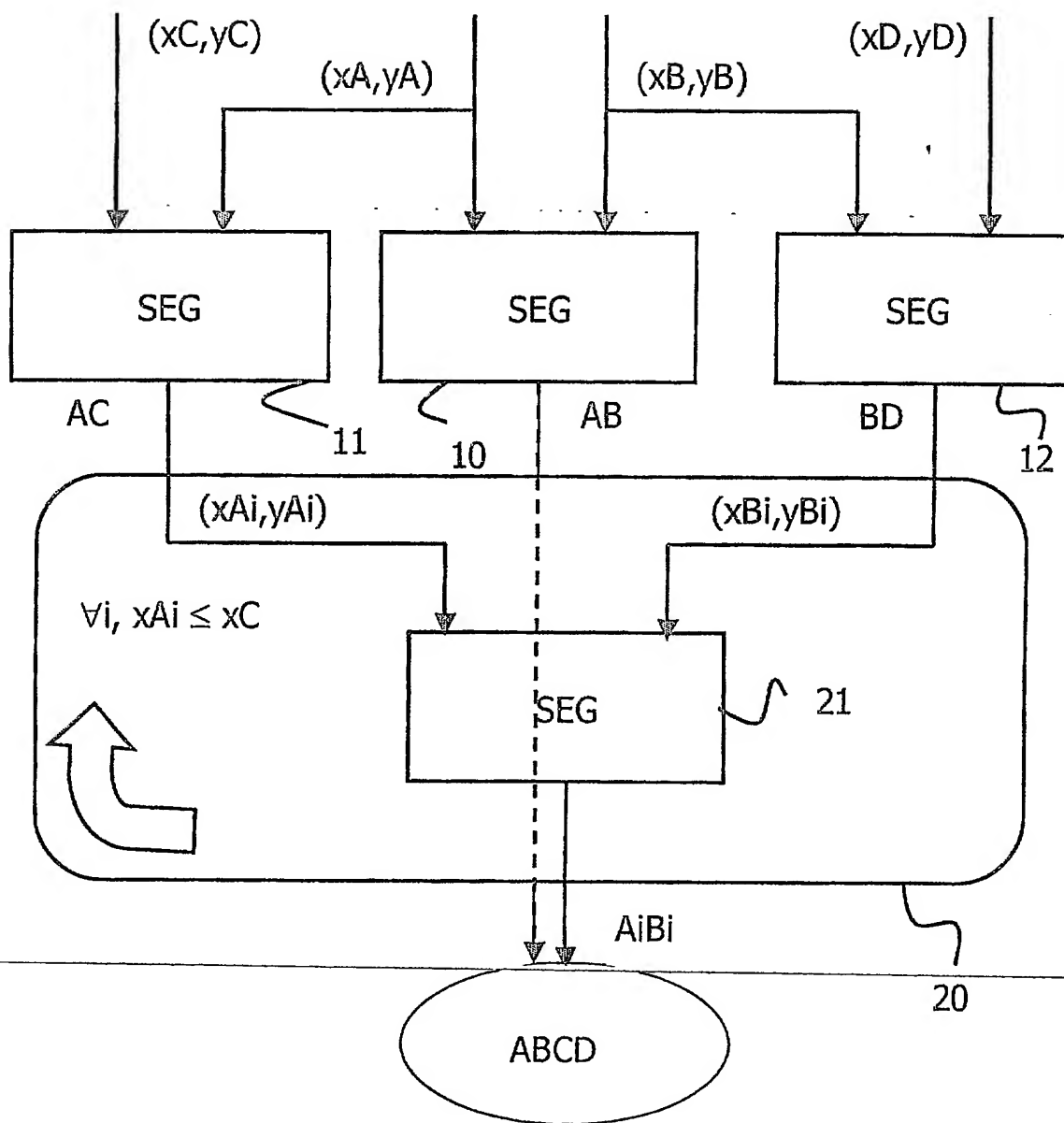


FIG. 3a



$$(x_{C2}, y_{C2}) = (x_i, y_i)$$

FIG. 3b

**FIG. 4**

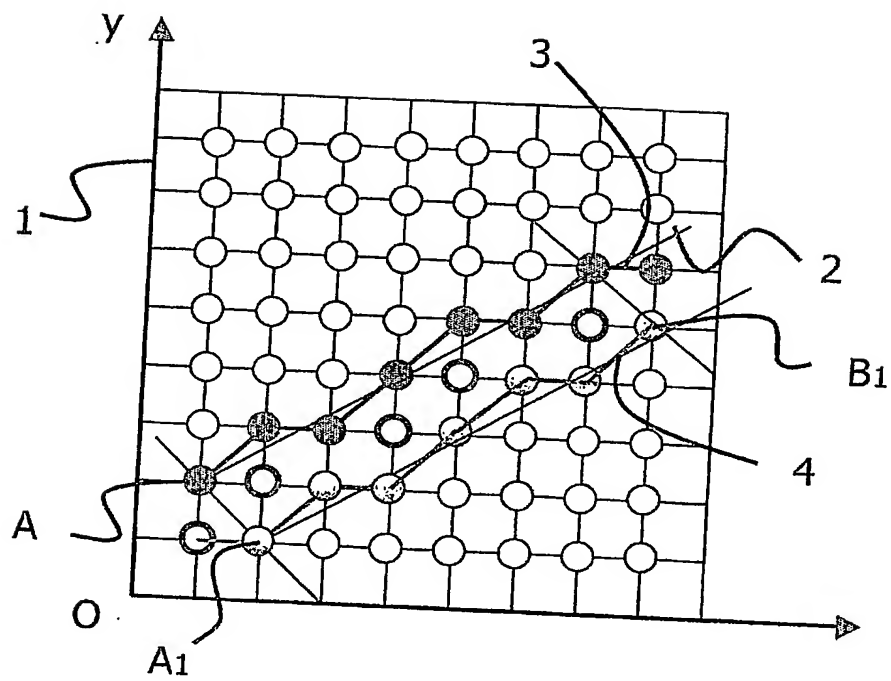


FIG. 5



6/8

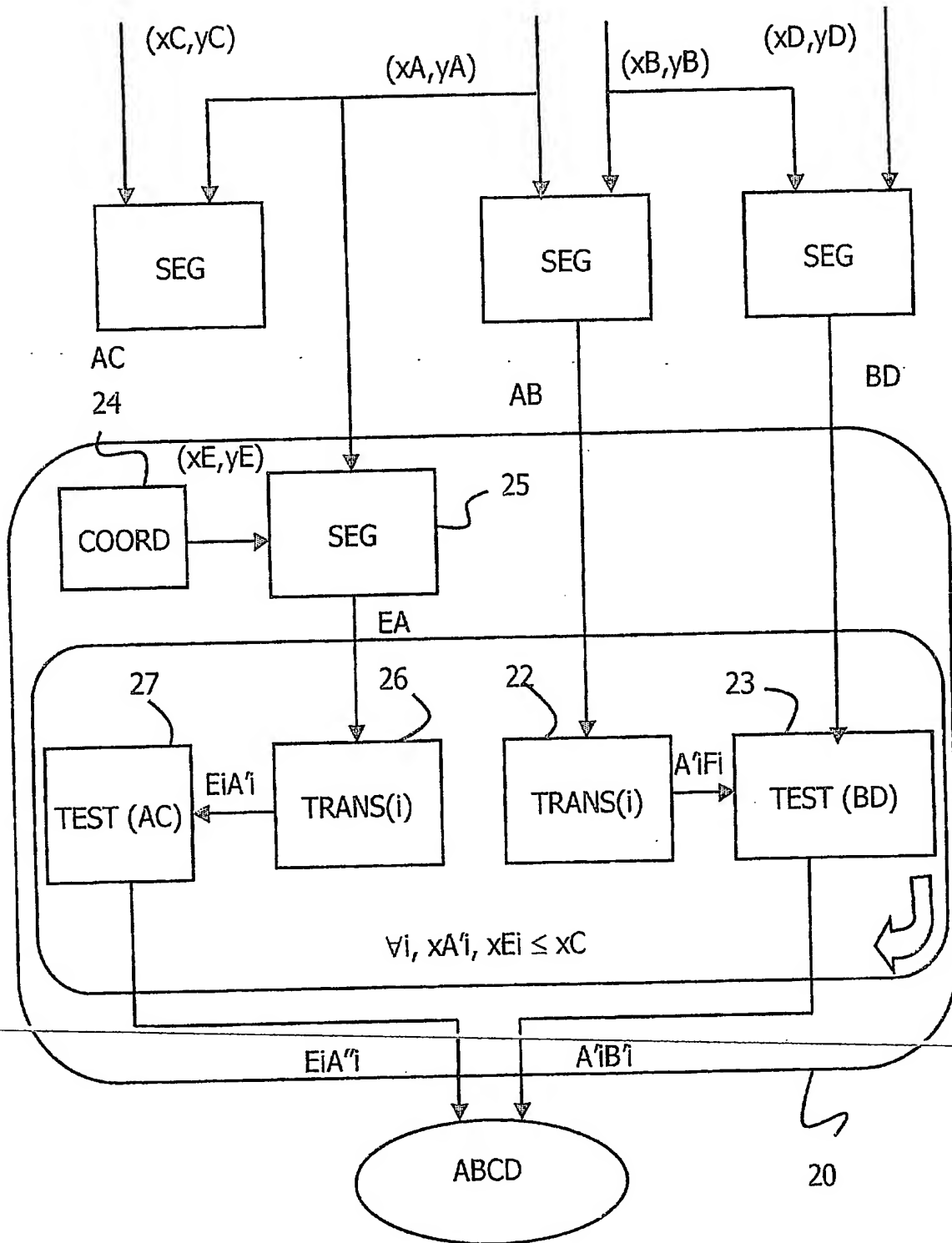


FIG. 6

7/8

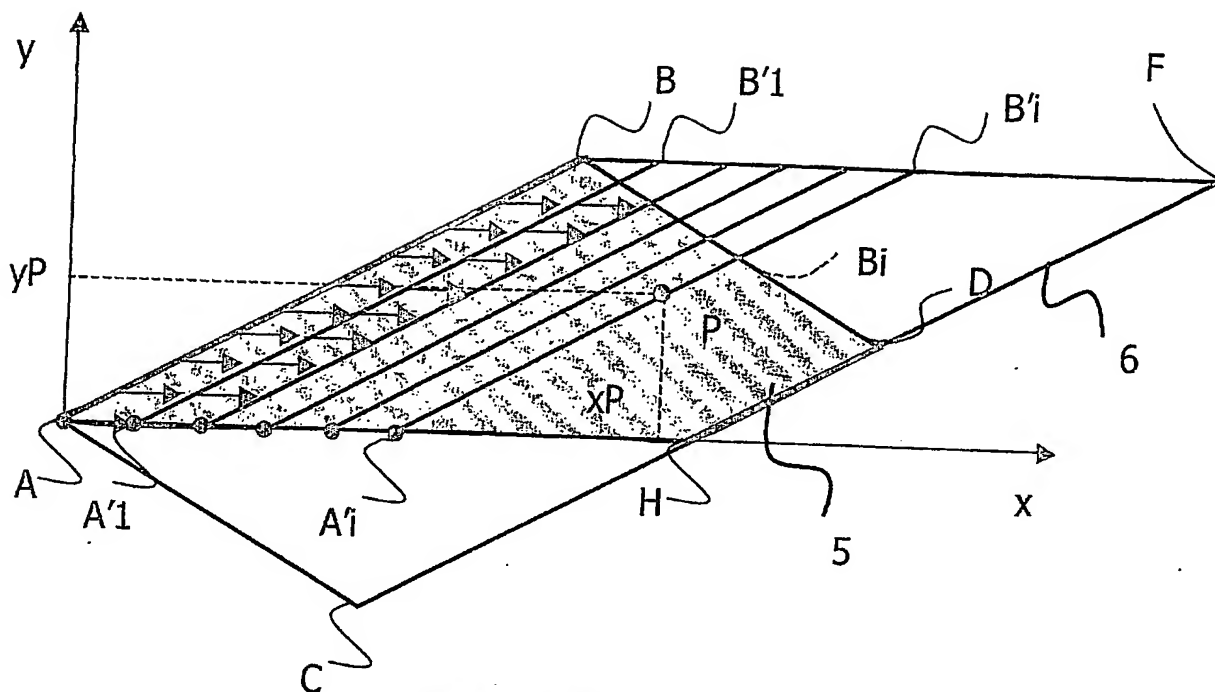


FIG. 7a

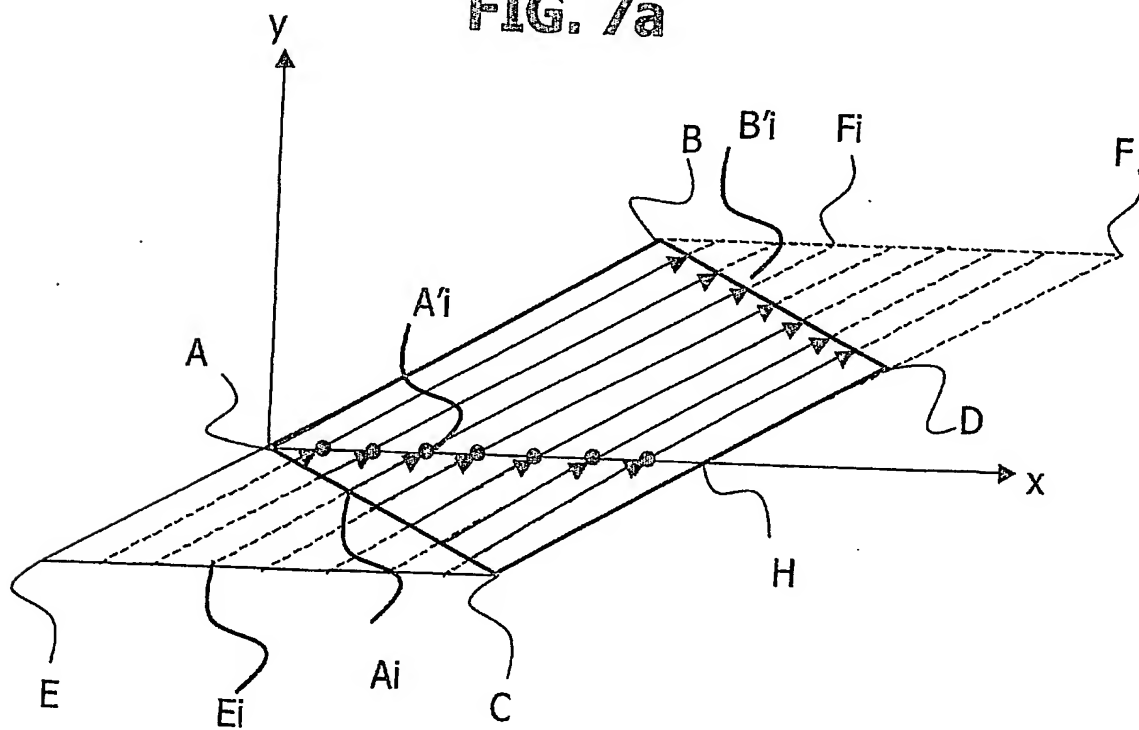


FIG. 7b

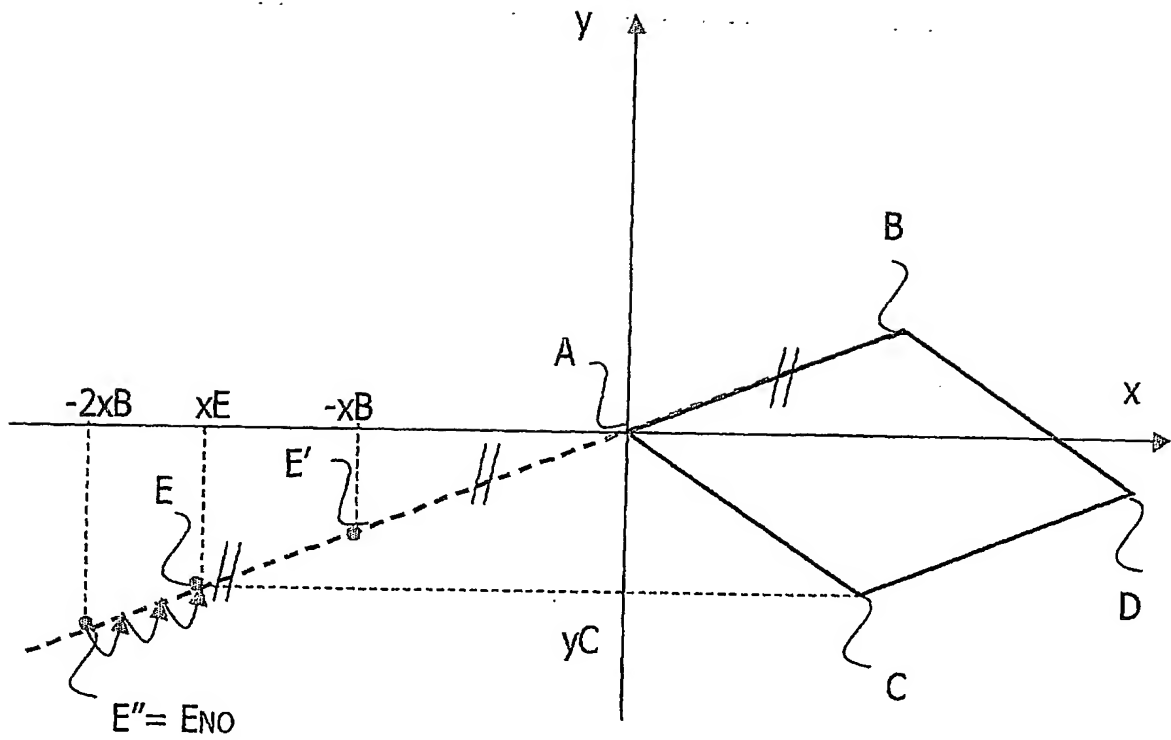


FIG. 8.



## BREVET D'INVENTION CERTIFICAT D'UTILITE

### Désignation de l'inventeur

Vos références pour ce dossier	PHFR030043
N°D'ENREGISTREMENT NATIONAL	
TITRE DE L'INVENTION	
	PROCÉDÉ ET SYSTÈME DE TRACE ET DE REMPLISSAGE D'UN PARALLELOGRAMME.
LE(S) DEMANDEUR(S) OU LE(S) MANDATAIRE(S):	
DESIGNE(NT) EN TANT QU'INVENTEUR(S):	
Inventeur 1	
Nom	ZHAO
Prénoms	Qin
Rue	156 Bd Haussmann
Code postal et ville	75008 PARIS
Société d'appartenance	S.P.I.D.
Inventeur 2	
Nom	PASQUIER
Prénoms	Laurent
Rue	156 Bd Haussmann
Code postal et ville	75008 PARIS
Société d'appartenance	S.P.I.D.
Inventeur 3	
Nom	DURANTON
Prénoms	Marc
Rue	156 Bd Haussmann
Code postal et ville	75008 PARIS
Société d'appartenance	S.P.I.D.

La loi n°78-17 du 6 janvier 1978 relative à l'informatique aux fichiers et aux libertés s'applique aux réponses faites à ce formulaire. Elle garantit un droit d'accès et de rectification pour les données vous concernant auprès de l'INPI.

42 2004 001209

PCT/IB2004/001209



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☒ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**